

CLAIM AMENDMENTS

1. (Amended) A method for supplying a consistent set of data to a software application, the method comprising the steps of:
- [launching said software application;]
- identifying a particular set of data that is required by the software application;
- [requesting a first] causing each process of a first set of two or more processes to obtain a snapshot time from a database server associated with a first database, wherein the snapshot times obtained by the first set of processes are associated with a same state of the particular set of data [time causes all subsequent reads of said first database by the first process to return data that reflects a database state associated with the snapshot time];
- after [the first] each process of the first set of processes obtains [the] a snapshot time, causing the first [process] set of processes to extract the particular set of data from the first database; [and]
- [supplying said software application with the particular set of data that was extracted from the first database.]
- causing the first set of processes to store a first copy of the particular set of data that is extracted from the first database in one or more files that are separate from the first database;
- causing a second set of one or more processes to store a second copy of the particular set of data that is stored in the one or more files, wherein the second copy is stored separate from the first database, and wherein no process in the second set of processes is in the first set of processes; and
- allowing the software application to access the second copy of the particular set of data.

2. (Amended) The method of claim 1, [further comprising the step of causing a second process to store the particular set of data] wherein the second copy is stored in a second database.

1 3. (Amended) The method of claim 2, wherein [the step of causing the second process to  
2 store the particular set of data in the second database includes the steps of]:  
3 [writing the particular set of data to] the one or more files include one or more flat files;  
4 and  
5 [executing a loader process,] the second set of processes is a set of one or more loader  
6 processes, wherein the set of loader [process] processes loads the second copy of  
7 the particular set of data into the second database based on the first copy that is  
8 stored in [from] the one or more flat files [to the second database].

1 4. (Amended) The method of claim 3, [wherein the step of writing the particular set of data  
2 to one or more flat files includes] further comprising the steps of:  
3 each process of the first set of [process] processes informing a coordinator process when  
4 [it] said each process has finished writing data to a particular flat file of the one or  
5 more flat files; and  
6 after each process of the first set of processes informs the coordinator process, the  
7 coordinator process [using the information to tell the] informing each loader  
8 process of the set of loader processes when [it] said each loader process can begin  
9 loading the second copy of the particular set of data into the second database  
10 based on each of the one or more flat [file] files [into the second database].

1 5. (Amended) The method of claim 3, wherein [the step of]:  
2 [writing the particular set of data to the flat file includes the step of writing the particular  
3 set of data to] the one or more flat files comprises a plurality of flat files; and  
4 [executing the loader process includes the step of executing] the set of loader processes  
5 comprises a plurality of loader processes, wherein the plurality of loader processes  
6 [load] loads the particular set of data from the plurality of flat files to the second  
7 database.

- 1 6. (Amended) The method of claim 2, [wherein the step of supplying said software  
2 application with data from said particular set of data includes] further comprising the  
3 steps of:  
4 said software application reading the second copy of the particular set of data stored in  
5 the second database; and  
6 said software application generating a planning schedule based on the second copy of the  
7 particular set of data.
- 1 7. (Not Amended) The method of claim 1, wherein the step of identifying the particular set  
2 of data includes the step of creating a copy table list, wherein the copy table list contains  
3 entries that identify the particular set of data in the first database.
- 1 8. (Amended) The method of claim 7, further comprising the steps of:  
2 executing a set of one or more delete [process] processes, wherein the set of delete  
3 [process] processes uses the copy table list to identify data that needs to be deleted  
4 in a second database; and  
5 deleting the identified data from the second database.
- 1 9. (Amended) The method of claim 7, [where] wherein the step of creating the copy table  
2 list includes the steps of:  
3 communicating with the software application to [identifying] identify a set of planning  
4 data, [where] wherein the planning data is required for generating a planning  
5 schedule; and  
6 creating the copy table list based on the identified set of planning data.
- 1 10. (Amended) The method of claim 1, wherein [the step of supplying said software  
2 application with data from said particular set of data includes the steps of:  
3 writing the particular set of data to] the one or more files include one or more flat files[;],  
4 and the method further comprises the steps of:

5 [supplying] allowing the software application to access the one or more flat files [to said  
6 software application, wherein] such that said software application [generates] can  
7 generate a planning schedule based on information contained in the one or more  
8 flat files.

1 11. (Twice Amended) A method for producing a copy of data from a first database, the  
2 method comprising the steps of:  
3 locking a first set of data in the first database;  
4 after locking the first set of data,  
5 requesting each process in a plurality of processes to obtain its own snapshot  
6 [times] time from a database server associated with said first database,  
7 wherein the snapshot times cause all subsequent reads of the first database  
8 by the plurality of processes to return data from the first database as of said  
9 snapshot times;  
10 waiting a particular period of time for the plurality of processes to be assigned  
11 snapshot times;  
12 releasing the locks on the first set of data in the first database;  
13 using a successful set of said plurality of processes to extract a copy of the first set  
14 of data from the first database, wherein:  
15 said successful set of said plurality of processes includes only those  
16 processes of the plurality of processes that were assigned a snapshot  
17 time within the particular period of time; and  
18 before the successful set of said plurality of processes finishes extracting  
19 the copy of the first set of data, the state of the first set of data is  
20 changed; and  
21 storing the copy of the first set of data separate from said first set of data.

1 12. (Not Amended) The method of claim 11, wherein the step of identifying the first set of  
2 data includes the step of creating a copy table list, wherein the copy table list contains  
3 entries that identify the first set of data in the first database.

1 13. (Amended) The method of claim 12, [where] wherein the step of creating the copy table  
2 list includes the steps of:  
3 identifying a set of planning data, [where] wherein the planning data is required to  
4 generate a planning schedule; and  
5 creating the copy table list based on the planning data required to generate the planning  
6 schedule.

1 14. (Amended) The method of claim 12, further comprising the steps of:  
2 executing a plurality of delete processes, wherein the plurality of delete processes [use]  
3 uses the copy table list to identify data that needs to be deleted in a second  
4 database; and  
5 deleting the identified data from the second database.

1 15. (Not Amended) The method of claim 11, wherein the step of storing the copy of the first  
2 set of data includes the steps of:  
3 writing the copy of the first set of data to a plurality of flat files; and  
4 executing a plurality of loader processes, wherein the plurality of loader processes load  
5 the copy of the first set of data from the plurality of flat files to a second database.

1 16. (Amended) The method of claim 15, wherein:  
2 the steps of writing the copy of the first set of data to [a] the plurality of flat files further  
3 includes the step of notifying a coordinator process that data has been written to  
4 one of the plurality of flat files; and  
5 the steps of executing the plurality of loader processes further includes the step of the  
6 coordinator, upon being notified that data has been written to one of the plurality  
7 of flat files, launching a loader process to load the first set of data from one of the  
8 plurality of flat files to the second database.

1 17. (Amended) The method of claim 15, wherein the step of writing the copy of the first set  
2 of data to the plurality of flat files includes the steps of:  
3 the plurality of [process] processes informing a coordinator process when [it] the plurality  
4 of processes has finished writing data to a particular flat file; and  
5 the coordinator process using the information to tell one of the plurality of loader  
6 processes when [it] said one of the plurality of loader processes can begin loading  
7 the particular flat file into the second database.

1 18. (Amended) The method of claim 11, wherein the step of requesting the plurality of  
2 processes to obtain [a] snapshot [time] times includes the step of requesting the plurality  
3 of processes based on a user input parameter, wherein the user input parameter identifies  
4 how many processes should be requested to obtain [a] snapshot [time] times.

1 19. (Amended) The method of claim 11, wherein the plurality of processes is a plurality of  
2 snapshot processes, and wherein the step of extracting the copy of the first set of data  
3 from the database includes the steps of:  
4 assigning a set of copy data to the plurality of snapshot processes; and  
5 retrieving data from the first database based on the set of copy data that was assigned to  
6 the plurality of snapshot processes.

1 20. (Amended) The method of claim 11, wherein the step of storing the copy of the first set  
2 of data includes the steps of storing the copy of the first set of data as blob files that are  
3 separate from said first set of data.

1 21. (Amended) The method of claim 11, wherein the step of storing the copy of the first set  
2 of data includes the steps of storing the copy of the first set of data in said first [of data]  
3 database.

22. (Amended) A computer-readable medium carrying one or more sequences of one or more instructions for supplying a consistent set of data to a software application, the one or more sequences of one or more instructions including instructions which, when executed by one or more processors, cause the one or more processors to perform the steps of: [launching said software application;]  
identifying a particular set of data that is required by the software application;  
[requesting a first] causing each process of a first set of two or more processes to obtain a snapshot time from a database server associated with a first database, wherein the snapshot times obtained by the first set of processes are associated with a same state of the particular set of data [time causes all subsequent reads of said first database by the first process to return data that reflects a database state associated with the snapshot time];  
after [the first] each process of the first set of processes obtains [the] a snapshot time, causing the first [process] set of processes to extract the particular set of data from the first database; [and]  
[supplying said software application with the particular set of data that was extracted from the first database.]  
causing the first set of processes to store a first copy of the particular set of data that is extracted from the first database in one or more files that are separate from the first database;  
causing a second set of one or more processes to store a second copy of the particular set of data that is stored in the one or more files, wherein the second copy is stored separate from the first database, and wherein no process in the second set of processes is in the first set of processes; and  
allowing the software application to access the second copy of the particular set of data.

23. (Amended) The computer-readable medium of claim 22, [wherein the computer-readable medium further comprises instructions for performing the step of causing a second process to store the particular set of data] wherein the second copy is stored in a second database.

1 24. (Amended) The computer-readable medium of claim 23, wherein [the step of causing the  
2 second process to store the particular set of data in the second database includes the steps  
3 of]:

4 [writing the particular set of data to] the one or more files include one or more flat files;  
5 and

6 [executing a loader process,] the second set of processes is a set of one or more loader  
7 processes, wherein the set of loader [process] processes loads the second copy of  
8 the particular set of data into the second database based on the first copy that is  
9 stored in [from] the one or more flat files [to the second database].

1 25. (Amended) The computer-readable medium of claim 24, [wherein the step of writing the  
2 particular set of data to one or more flat files includes] further comprising instructions for  
3 performing the steps of:

4 each process of the first set of [process] processes informing a coordinator process when  
5 [it] said each process has finished writing data to a particular flat file of the one or  
6 more flat files; and

7 after each process of the first set of processes informs the coordinator process, the  
8 coordinator process [using the information to tell the] informing each loader  
9 process of the set of loader processes when [it] said each loader process can begin  
10 loading the second copy of the particular set of data into the second database  
11 based on each of the one or more flat [file] files [into the second database].

1 26. (Not Amended) The computer-readable medium of claim 22, wherein the step of  
2 identifying the particular set of data includes the step of creating a copy table list,  
3 wherein the copy table list contains entries that identify the particular set of data in the  
4 first database.



1 27. (Twice Amended) A computer-readable medium carrying one or more sequences of one  
2 or more instructions for producing a copy of data from a first database, the one or more  
3 sequences of one or more instructions including instructions which, when executed by  
4 one or more processors, cause the one or more processors to perform the steps of:  
5 locking a first set of data in the first database;  
6 after locking the first set of data,  
7 requesting each process in a plurality of processes to obtain its own snapshot  
8 [times] time from a database server associated with said first database,  
9 wherein the snapshot times cause all subsequent reads of the first database  
10 by the plurality of processes to return data from the first database as of said  
11 snapshot times;  
12 waiting a particular period of time for the plurality of processes to be assigned  
13 snapshot times;  
14 releasing the locks on the first set of data in the first database;  
15 using a successful set of said plurality of processes to extract a copy of the first set  
16 of data from the first database, wherein:  
17 said successful set of said plurality of processes includes only those  
18 processes of the plurality of processes that were assigned a snapshot  
19 time within the particular period of time; and  
20 before the successful set of said plurality of processes finishes extracting  
21 the copy of the first set of data, the state of the first set of data is  
22 changed; and  
23 storing the copy of the first set of data separate from said first set of data.

1 28. (Not Amended) The computer-readable medium of claim 27, wherein the step of  
2 identifying the first set of data includes the step of creating a copy table list, wherein the  
3 copy table list contains entries that identify the first set of data in the first database.

1 29. (Not Amended) The computer-readable medium of claim 27, wherein the step of storing  
2 the copy of the first set of data includes the steps of:  
3 writing the copy of the first set of data to a plurality of flat files; and  
4 executing a plurality of loader processes, wherein the plurality of loader processes load  
5 the copy of the first set of data from the plurality of flat files to a second database.

1 30. (Amended) The computer-readable medium of claim 29, wherein the step of writing the  
2 copy of the first set of data to the plurality of flat files includes the steps of:  
3 the plurality of [process] processes informing a coordinator process when [it] the plurality  
4 of processes has finished writing data to a particular flat file; and  
5 the coordinator process using the information to tell one of the plurality of loader  
6 processes when [it] said one of the plurality of loader processes can begin loading  
7 the particular flat file into the second database.

1 31. (Amended) The computer-readable medium of claim 27, wherein the plurality of  
2 processes is a plurality of snapshot processes, and wherein the step of extracting the copy  
3 of the first set of data from the database includes the steps of:  
4 assigning a set of copy data to the plurality of snapshot processes; and  
5 retrieving data from the first database based on the set of copy data that was assigned to  
6 the plurality of snapshot processes.

1 32. (Amended) A computer system for supplying a consistent set of data to a software  
2 application, the computer system comprising:  
3 a memory;  
4 one or more processors coupled to the memory; and  
5 a set of computer instructions contained in the memory, the set of computer instructions  
6 including computer instructions which when executed by the one or more  
7 processors, cause the one or more processors to perform the steps of:  
8 [launching said software application;]  
9 identifying a particular set of data that is required by the software application;

[requesting a first] causing each process of a first set of two or more processes to  
obtain a snapshot time from a database server associated with a first  
database, wherein the snapshot times obtained by the first set of processes  
are associated with a same state of the particular set of data [time causes  
all subsequent reads of said first database by the first process to return data  
that reflects a database state associated with the snapshot time];  
after [the first] each process of the first set of processes obtains [the] a snapshot  
time, causing the first [process] set of processes to extract the particular set  
of data from the first database; [and]  
[supplying said software application with the particular set of data that was  
extracted from the first database.]  
causing the first set of processes to store a first copy of the particular set of data  
that is extracted from the first database in one or more files that are  
separate from the first database;  
causing a second set of one or more processes to store a second copy of the  
particular set of data that is stored in the one or more files, wherein the  
second copy is stored separate from the first database, and wherein no  
process in the second set of processes is in the first set of processes; and  
allowing the software application to access the second copy of the particular set of  
data.

33. (Amended) The computer system of claim 32, [further including instructions for  
performing the step of causing a second process to store the particular set of data]  
wherein the second copy is stored in a second database.

34. (Amended) The computer system of claim 33, wherein [the step of causing the second  
process to store the particular set of data in the second database includes the steps of]:  
[writing the particular set of data to] the one or more files include one or more flat files;  
and

5 [executing a loader process,] the second set of processes is a set of one or more loader  
6 processes, wherein the set of loader [process] processes loads the second copy of  
7 the particular set of data into the second database based on the first copy that is  
8 stored in [from] the one or more flat files [to the second database].

1 35. (Amended) The computer system of claim 34, [wherein the step of writing the particular  
2 set of data to one or more flat files includes] further comprising instructions for  
3 performing the steps of:  
4 each process of the first set of [process] processes informing a coordinator process when  
5 [it] said each process has finished writing data to a particular flat file of the one or  
6 more flat files; and  
7 after each process of the first set of processes informs the coordinator process, the  
8 coordinator process [using the information to tell the] informing each loader  
9 process of the set of loader processes when [it] said each loader process can begin  
10 loading the second copy of the particular set of data into the second database  
11 based on each of the one or more flat [file] files [into the second database].

1 36. (Not Amended) The computer system of claim 32, wherein the step of identifying the  
2 particular set of data includes the step of creating a copy table list, wherein the copy table  
3 list contains entries that identify the particular set of data in the first database.

1 37. (Twice Amended) A computer system for producing a copy of data from a first database,  
2 the computer system comprising:  
3 a memory;  
4 one or more processors coupled to the memory; and  
5 a set of computer instructions contained in the memory, the set of computer instructions  
6 including computer instructions which when executed by the one or more  
7 processors, cause the one or more processors to perform the steps of:  
8 locking a first set of data in the first database;  
9 after locking the first set of data,

10 requesting each process in a plurality of processes to obtain its own  
11 snapshot [times] time from a database server associated with said  
12 first database, wherein the snapshot times cause all subsequent reads  
13 of the first database by the plurality of processes to return data from  
14 the first database as of said snapshot times;  
15 waiting a particular period of time for the plurality of processes to be  
16 assigned snapshot times;  
17 releasing the locks on the first set of data in the first database;  
18 using a successful set of said plurality of processes to extract a copy of the  
19 first set of data from the first database, wherein:  
20 said successful set of said plurality of processes includes only those  
21 processes of the plurality of processes that were assigned a  
22 snapshot time within the particular period of time; and  
23 before the successful set of said plurality of processes finishes  
24 extracting the copy of the first set of data, the state of the first  
25 set of data is changed; and  
26 storing the copy of the first set of data separate from said first set of data.

1 38. (Not Amended) The computer system of claim 37, wherein the step of identifying the  
2 first set of data includes the step of creating a copy table list, wherein the copy table list  
3 contains entries that identify the first set of data in the first database.

1 39. (Not Amended) The computer system of claim 37, wherein the step of storing the copy  
2 of the first set of data includes the steps of:  
3 writing the copy of the first set of data to a plurality of flat files; and  
4 executing a plurality of loader processes, wherein the plurality of loader processes load  
5 the copy of the first set of data from the plurality of flat files to a second database.

1 40. (Amended) The computer system of claim 39, wherein the step of writing the copy of the  
2 first set of data to the plurality of flat files includes the steps of:  
3 the plurality of [process] processes informing a coordinator process when [it] the plurality  
4 of processes has finished writing data to a particular flat file; and  
5 the coordinator process using the information to tell one of the plurality of loader  
6 processes when [it] said one of the plurality of loader processes can begin loading  
7 the particular flat file into the second database.

1 41. (Amended) The computer system of claim 37, wherein the plurality of processes is a  
2 plurality of snapshot processes, and wherein the step of extracting the copy of the first set  
3 of data from the database includes the steps of:  
4 assigning a set of copy data to the plurality of snapshot processes; and  
5 retrieving data from the first database based on the set of copy data that was assigned to  
6 the plurality of snapshot processes.

1 42. (New, Amended) A method for extracting data from a first database, the method  
2 comprising the steps of:  
3 causing each process in a set of two or more processes to obtain its own snapshot time  
4 that cause all subsequent reads of the first database by each process in the set of  
5 processes to return data from a same state of at least a portion of the first database  
6 as will be returned to all other processes in the set of processes;  
7 causing the set of processes to extract a copy of the portion of the first database; and  
8 wherein before the set of processes finish extracting said copy, the state of the portion of  
9 the first database is changed.

1 43. (New) The method of claim 42, wherein the set of processes are included in a plurality of  
2 processes, and the step of causing the set of processes to obtain snapshot times comprises  
3 the step of:  
4 causing the plurality of processes to attempt to obtain snapshot times that cause all  
5 subsequent reads of the first database by each process of the plurality of processes  
6 that successfully obtains one of the snapshot times to return data from the same  
7 state of at least the portion of the first database as will be returned to all other  
8 processes of the plurality of processes that successfully obtain one of the snapshot  
9 times.

1 44. (New) The method of claim 43, wherein at least one process of the plurality of processes  
2 fails to one of the snapshot times within a particular period of time.

1 45. (New) The method of claim 42, wherein each process of the set of processes obtains a  
2 snapshot time by sending a request for a snapshot time to a database management system  
3 that is associated with the first database and that assigns a snapshot time in response to  
4 the request.

1 46. (New) The method of claim 42, wherein each process of the set of processes is assigned  
2 to extract a particular subset of the portion from the first database.

1 47. (New) The method of claim 42, wherein each process of the set of processes obtains a  
2 snapshot time within a particular period of time.

1 48. (New) The method of claim 47, wherein the set of processes does not begin to extract the  
2 copy of the portion until after expiration of the particular period of time.

1 49. (New) The method of claim 47, wherein at least one process of the set of processes  
2 begins to extract the copy of the portion prior to expiration of the particular period of  
3 time.

1 50. (New) The method of claim 47, further comprising the step of:

2 locking the portion of the first database during the particular period of time.

1 51. (New) The method of claim 42, further comprising the step of:

2 using a set transaction read only command such that each process of the set of processes  
3 can obtain a snapshot time.

1 52. (New) The method of claim 42, further comprising the step of:

2 storing the copy separate from the portion.

1 53. (New) The method of claim 52, wherein the step of storing the copy comprises the step  
2 of:

3 storing the copy in a second database.

1 54. (New) The method of claim 53, wherein step of storing the copy in the second database  
2 comprises the step of:

3 storing the copy as one or more large binary objects in the second database.

1 55. (New) The method of claim 52, wherein the step of storing the copy comprises the step  
2 of:

3 storing the copy in the first database.

1 56. (New) The method of claim 55, wherein step of storing the copy in the first database  
2 comprises the step of:

3 storing the copy as one or more large binary objects in the first database.

1 57. (New) The method of claim 42, wherein the copy is a first copy and the set of two or  
2 more processes is a first set of two or more processes, and wherein the method further  
3 comprises the steps of:

4 causing the first copy to be stored into one or more files by the first set of processes; and



5       requesting each process of a second set of one or more processes to load a second copy  
6       into a second database, wherein the second copy is based on the first copy stored  
7       in the one or more files.

1   58. (New) The method of claim 57, wherein:

2       the one or more files include one or more flat files;  
3       the first set of two or more processes includes a set of two or more snapshot worker  
4       processes;  
5       the second set of one or more processes includes a set of one or more loader worker  
6       processes; and  
7       the second database is a snapshot table database.

1   59. (New) The method of claim 57, further comprising the steps of:

2       each process of the first set of processes informing a coordinator process when said each  
3       process of the first set of processes has finished writing to a particular file of the  
4       one or more files; and  
5       after each process of the first set of processes informs the coordinator process, the  
6       coordinator process informing each process of the second set of processes when  
7       said each process of the second set of processes can begin loading the second copy  
8       to the second database.

1   60. (New) The method of claim 57, further comprising the steps of:

2       allowing an application to read the second copy that is stored in the second database,  
3       wherein the application generates a schedule based on data within the second  
4       copy.

1   61. (New) The method of claim 42, further comprising the steps of:

2       causing the first copy to be stored into one or more files by the set of processes; and  
3       allowing an application to read the first copy that is stored in the one or more files,  
4       wherein the application generates a schedule based on data within the first copy.

1 62. (New) The method of claim 42, further comprising the step of:  
2 using a list of database tables to identify the portion of the first database.

1 63. (New) The method of claim 62, wherein the set of processes uses the list of database  
2 tables to extract the copy from the first database.

1 64. (New) The method of claim 62, wherein the list of database tables is a copy table list that  
2 is generated by one process of the set of processes, and wherein the method further  
3 comprises the steps of:  
4 requesting each process of a third set of processes to identify and delete, based on the  
5 copy table list, data from a second database.

1 65. (New) The method of claim 62, further comprising the steps of:  
2 communicating with an application to identify a set of data for generating a schedule; and  
3 causing the list of database tables to be created based on the identified set of data.

1 66. (New, Amended) A computer-readable medium carrying one or more sequences of one  
2 or more instructions for extracting data from a first database, the one or more sequences  
3 of one or more instructions including instructions which, when executed by one or more  
4 processors, cause the one or more processors to perform the steps of:  
5 causing each processor a set of two or more processes to obtain its own snapshot time that  
6 cause all subsequent reads of the first database by each process in the set of  
7 processes to return data from a same state of at least a portion of the first database  
8 as will be returned to all other processes in the set of processes;  
9 causing the set of processes to extract a copy of the portion of the first database; and  
10 wherein before the set of processes finish extracting said copy, the state of the portion of  
11 the first database is changed.

1 67. (New) The computer-readable medium of claim 66, wherein the set of processes are  
2 included in a plurality of processes, and the step of causing the set of processes to obtain  
3 snapshot times comprises the step of:  
4 causing the plurality of processes to attempt to obtain snapshot times that cause all  
5 subsequent reads of the first database by each process of the plurality of processes  
6 that successfully obtains one of the snapshot times to return data from the same  
7 state of at least the portion of the first database as will be returned to all other  
8 processes of the plurality of processes that successfully obtain one of the snapshot  
9 times.

1 68. (New) The computer-readable medium of claim 66, wherein at least one process of the  
2 plurality of processes fails to one of the snapshot times within a particular period of time.

1 69. (New) The computer-readable medium of claim 66, wherein each process of the set of  
2 processes obtains a snapshot time by sending a request for a snapshot time to a database  
3 management system that is associated with the first database and that assigns a snapshot  
4 time in response to the request.

1 70. (New) The computer-readable medium of claim 66, wherein each process of the set of  
2 processes is assigned to extract a particular subset of the portion from the first database.

1 71. (New) The computer-readable medium of claim 66, wherein each process of the set of  
2 processes obtains a snapshot time within a particular period of time.

1 72. (New) The computer-readable medium of claim 71, wherein the set of processes does not  
2 begin to extract the copy of the portion until after expiration of the particular period of  
3 time.

1 73. (New) The computer-readable medium of claim 71, wherein at least one process of the  
2 set of processes begins to extract the copy of the portion prior to expiration of the  
3 particular period of time.

1 74. (New) The computer-readable medium of claim 66, further comprising instructions for  
2 performing the step of:  
3 locking the portion of the first database during the particular period of time.

1 75. (New) The computer-readable medium of claim 66, further comprising instructions for  
2 performing the step of:  
3 using a set transaction read only command such that each process of the set of processes  
4 can obtain a snapshot time.

1 76. (New) The computer-readable medium of claim 66, further comprising instructions for  
2 performing the step of:  
3 storing the copy separate from the portion.

1 77. (New) The computer-readable medium of claim 76, wherein the step of storing the copy  
2 comprises the step of:  
3 storing the copy in a second database.

1 78. (New) The computer-readable medium of claim 77, wherein step of storing the copy in  
2 the second database comprises the step of:  
3 storing the copy as one or more large binary objects in the second database.

1 79. (New) The computer-readable medium of claim 76, wherein the step of storing the copy  
2 comprises the step of:  
3 storing the copy in the first database.

1 80. (New) The computer-readable medium of claim 79, wherein step of storing the copy in  
2 the first database comprises the step of:  
3 storing the copy as one or more large binary objects in the first database.

1 81. (New) The computer-readable medium of claim 66, wherein the copy is a first copy and  
2 the set of two or more processes is a first set of two or more processes, and further  
3 comprising instructions for performing the steps of:  
4 causing the first copy to be stored into one or more files by the first set of processes; and  
5 requesting each process of a second set of one or more processes to load a second copy  
6 into a second database, wherein the second copy is based on the first copy stored  
7 in the one or more files.

1 82. (New) The computer-readable medium of claim 81, wherein:  
2 the one or more files include one or more flat files;  
3 the first set of two or more processes includes a set of two or more snapshot worker  
4 processes;  
5 the second set of one or more processes includes a set of one or more loader worker  
6 processes; and  
7 the second database is a snapshot table database.

1 83. (New) The computer-readable medium of claim 81, further comprising instructions for  
2 performing the steps of:  
3 each process of the first set of processes informing a coordinator process when said each  
4 process of the first set of processes has finished writing to a particular file of the  
5 one or more files; and  
6 after each process of the first set of processes informs the coordinator process, the  
7 coordinator process informing each process of the second set of processes when  
8 said each process of the second set of processes can begin loading the second copy  
9 to the second database.

1 84. (New) The computer-readable medium of claim 81, further comprising instructions for  
2 performing the step of:  
3 allowing an application to read the second copy that is stored in the second database,  
4 wherein the application generates a schedule based on data within the second  
5 copy.

1 85. (New) The computer-readable medium of claim 66, further comprising instructions for  
2 performing the steps of:  
3 causing the first copy to be stored into one or more files by the set of processes; and  
4 allowing an application to read the first copy that is stored in the one or more files,  
5 wherein the application generates a schedule based on data within the first copy.

1 86. (New) The computer-readable medium of claim 66, further comprising instructions for  
2 performing the step of:  
3 using a list of database tables to identify the portion of the first database.

1 87. (New) The computer-readable medium of claim 86, wherein the set of processes uses the  
2 list of database tables to extract the copy from the first database.

1 88. (New) The computer-readable medium of claim 86, wherein the list of database tables is  
2 a copy table list that is generated by one process of the set of processes, and further  
3 comprising instructions for performing the step of:  
4 requesting each process of a third set of processes to identify and delete, based on the  
5 copy table list, data from a second database.

1 89. (New) The computer-readable medium of claim 86, further comprising instructions for  
2 performing the steps of:  
3 communicating with an application to identify a set of data for generating a schedule; and  
4 causing the list of database tables to be created based on the identified set of data.

1 90. (New) The method of claim 1, wherein the set of two or more processes includes at least  
2 two processes that obtain different shapshot times.

1 91. (New) The method of claim 11, wherein the plurality of processes includes at least two  
2 processes that obtain different shapshot times.

1 92. (New) The computer-readable medium of claim 22, wherein the two or more processes  
2 includes at least two processes that obtain different shapshot times.

1 93. (New) The computer-readable medium of claim 27, wherein the plurality of processes  
2 includes at least two processes that obtain different shapshot times.

1 94. (New) The computer system of claim 32, wherein the set of two or more processes  
2 includes at least two processes that obtain different shapshot times.

1 95. (New) The computer system of claim 37, wherein the plurality of processes includes at  
2 least two processes that obtain different shapshot times.

1 96. (New) The method of claim 42, wherein the set of two or more processes includes at  
2 least two processes that obtain different shapshot times.

- 1   97. (New) The computer-readable medium of claim 66, wherein the set of two or more  
2       processes includes at least two processes that obtain different shapshot times.